Intelligent Traffic Management and Load Balance Based on Spike ISDN-IoT

Nadia Adnan Shiltagh Al-Jamali,*Member, IEEE,*and Hamed S. Al-Raweshidy,*Senior, IEEE*

***Abstract*—An Intelligent Software Defined Network (ISDN) based on an intelligent controller, can manage and control the network in a remarkable way. In this paper, a methodology is proposed to estimate the packet flow at the sensing plane in the Software Defined Network-Internet of Things (SDN-IoT) based on a Partial Recurrent Spike Neural Network (PRSNN) congestion controller, to predict the next step ahead of packet flow and thus, reduce the congestion that may occur. That is, the proposed model (Spike ISDN-IoT) is enhanced with a congestion controller. This controller works as a proactive controller in the proposed model. In addition, we propose another intelligent clustering controller based on an artificial neural network, which operates as a reactive controller, to manage the clustering in the sensing area of the Spike ISDN-IoT. Hence, an intelligent queuing model is introduced to manage the flow table buffer capacity of the spike ISDN-IoT network, such that the Quality of Service (QoS) of the whole network is improved. A modified training algorithm is introduced to train the PRSNN to adjust its weight and threshold. The simulation results demonstrate that the QoS is improved by (14.36%) when using the proposed model as compared with a convolutional neural network (CNN).**

***Index Terms*—Partial Recurrent Spike NN, cluster head, SDN-IoT, traffic load prediction, Quality of Service.**

I. Introduction

THE concept of the Internet of Things (IoT) has been made a reality by the creation of Wireless Sensor Networks (WSNs), which have the capability of monitoring or controlling different applications across the connectivity of the Internet. The basic idea of IoT is to enable real objects that are inserted with sensors, actuators, and network connectivity to accumulate and shuffle data among themselves in a cooperative way [1]. In other words, the IoT can be described by this formula (Things + Intelligence + Network = IoT) [2]. Many applications in the field of networks and the Internet require high speed, accuracy, security, and a high quality of services in the transfer of data. Accordingly, many solutions to enhance the Internet and computer networks with a high quality of services have been proposed, one of which is SDN-IoT. In an SDN, the data plane basically consists of a number of switches, routers, and gateways, while the control plane is responsible for taking the decisions for each node in the data plane using a southbound interface. [3]. The SDN controller has two interfaces: southbound and northbound. The role of the southbound interface has been described above, while the northbound one is tasked with providing services in the form of applications on the top of the SDN controller [4]. The proficient protocol that enables the controller in the SDN network to reach the switches and routers in the data plane is referred to as OpenFlow [5]. This has been adopted in a wide range of SDN applications such as Wide Area Networks (WAN), Internet exchange point, data center networks and cellular networks [6].

Manuscript received November 20,2019.(Corresponding author: Nadia A. Al-Jamali)

Nadia Adnan Shiltagh Al-Jamli with the department of Computer Engineering, University of Baghdad, Baghdad, Iraq. and with the department of Electronic and Computer Engineering, Brunel University London, London UB8 3PH, U.K.(e-mail: nadiaadnanshiltagh.aljamli@brunel.ac.uk).

Hamed S. Al-Raweshidy is with the department of Electronic and Computer Engineering, Brunel University London, London UB8 3PH, U.K. (e-mail: hamed.al-raweshidy@brunel.ac.uk). .

A. Motivation

The amount of data flow in the data plane is the most important issue in the field of traffic management and load balance in SDN networks. As the number of sensing devices that communicate with the switches in data plane is increased, the traffic load in the queuing buffer of the SDN-IoT gateway will also be increased. Also, as the number of switches in an SDN increases, the performance of the centralized controller in its control plane will fail to process all the requests coming from the switches. The use of artificial intelligent networks and machine learning with SDN has received increasingly marked interest in recent years. [7] gives an overview of machine learning algorithms that have been applied in the realm of SDN,which is providing novel opportunities to interleave intelligence in networks. The offerings of SDN, e.g., a control layer with comprehensive control of the network, the dynamic updating of the flow table entities and traffic analysis, can be strengthened further by applying intelligent techniques with it [7]. Combined with SDN, Artificial Intelligence (AI) can provide solutions to network problems based on classification and estimation techniques [8]. Intelligent traffic prediction is an important issue in SDN-IoT. Deep learning based on an artificial neural network (ANN) has demonstrated its proficiency in traffic management, load balance and routing in SDN networks [9]–[14]. One crucial requirement for improving network performance is optimizing the routing process of SDN, while maintaining the QoS [14]. The traditional SDN implementation based on a logically centralized controller has several constraints, including poor scalability and unreliable performance. With the fast growth of Internet flow and scale, this means that network sensor devices are widely spread, but the network range that a single controller can support is limited. In order to address the problem of low network performance and single point malfunction caused by exceeding traffic for a single controller, multiple controllers are usually implemented in the network, thereby delivering distributed

control management. With this arrangement, the control plane is split into several sub realms, with each controller only needing to manage the switches in its own. This can alleviate the deficiencies of the control plane in terms of reliability, scalability and versatility [15].
The design of an intelligent controller based on AI is the main topic in this paper. However, it is deemed appropriate to choose an algorithm that is more biologically realistic than an ANN. Spiking Neural Networks (SNNs) the "third generation of ANNs" are so and arguably the only viable option, if the aim is to gain clear insights into how the brain computes. Moreover, SNNs are more hardware friendly and energy-effective than ANNs [16]. SNNs are dynamic systems, with time being a more important factor than for conventional feedforward ANNs [17].

B. Contributions

This paper introduces a Partial Recurrent Spiking Neural Network (PRSNN) as a congestion controller in the proposed model. The PRSNN is a type of SNN with partial feedback in the hidden layer. Also, another controller based on ANN is introduced to manage the sensors in the spike ISDN-IoT network.

The main contributions of this paper can be summarized as follows:

1.We propose a spike ISDN-IoT model with two intelligent controllers in SDN intelligent stack, both of which are placed in the SDN control plane. One of them, which is based on PRSNN, estimates the amount of packet flow in the network, whilst the other, which is based on an ANN controller, selects and manages the cluster head of the sensors in the sensing area.

2. We propose an intelligent queuing model to estimate the capacity of the buffer size in the spike ISDN-IoT network based on a PRSNN controller.

3. We propose a modified training algorithm for PRSNN to update its weights, the delay and the threshold values.

The remainder of this paper is organized as follows. Section II reviews related works, section III presents the proposed system model with the network architecture and section IV presents the modified training algorithm. Then, in section V, the evaluation setup is presented and in section VI the results are shown, with the QoS improvements being discussed. Finally, in section VII the conclusion to the paper is provided.

II. Related Research Work

This section introduces the most recent research relating to the use of deep learning in traffic management and load balance applications in SDN networks. Mao *et al.* [14] proposed a non-supervised deep learning convolutional neural network (CNN) based routing methodology for a Software Defined Wireless Network, which can control the traffic of the network better than conventional routing protocols, with higher service quality.Tang *et al.* [9] proposed two deep-learning CNNs based on intelligent partial overlapping channel assignment to route traffic in a wireless SDN-IoT network, which improves the performance of the network. they utilized deep learning to predict the future traffic loads of switches.

Tang *et al.* [12] proposed a deep learning CNN based traffic load prediction algorithm for predicting traffic load at the next time interval and preventing congestion in an SDN-IoT network, which significantly outperforms the conventional method. Mao *et al.* [13] proposed intelligent routing based on a real-time deep learning strategy for a CNN in an SDN communication system. Yu *et al.* [10] suggested a deep reinforcement learning mechanism for an SDN to optimize the routing of the sensing area, which provides good convergence and effective routing services. Kumar and Vidyarthi [18] proposed a green routing algorithm based on particle swarm optimization for optimizing the number of control nodes and their clustering. The results obtained indicate a significant extension of the lifetime of the sensor network. Lin and Tsai [19] proposed a controller system for enhancing network scalability and reducing computation delay in SDNs, whilst meeting QoS requirements based on hierarchical edge-cloud SDN (HECSDN). Xu *et al.* [20] showed that multiple distributed controllers can be used in SDNs to improve scalability and reliability, where each manages one static partition of the network. The concept of Software Defined Wireless Sensor Network is experiencing rapid growth in the domain of IoT. The SDSense is a novel architecture proposed in [21], which entails an SDN based WSN design, where software enabled sensors are dynamically reconfigured to adapt to current network conditions, which significantly improves network performance. Misra *et al.* [22] proposed a situation-aware protocol switching scheme for software defined wireless sensor networks to support application in real-time.They showed that their protocol is capable of enhancing the network performance. Dias *et al.* [23] designed and implemented a scalable system architecture that integrates a WSN into IoT. Priority-based virtual machine allocation and a network traffic management scheme with bandwidth allocation along with a dynamic flow pathing mechanism were proposed by Son and Buyya [24]. Al-Shammari *et al.* [25] proposed a traffic flow management policy to allocate and organize traffic flow network resources.

AI has become a very important issue and researchers have been devising procedures for improving this area in the field of training algorithms, where SNNs are proving to be remarkably effective. There are many algorithms that have been proposed and implemented for training an SNN [17], [26]–[30].

Different from the reviewed literature, this paper implements two intelligent controllers in the spike ISDN-IoT control plane based on SDN intelligent stack. Also, we present a modified training algorithm to enhance the controllability of a spike ISDN-IoT network. The modification of the training algorithm is based on the spike back propagation (SBP) [26], [30]. Our proposed algorithm introduces a further training mechanism to prevent the occurrence of unwanted spikes that may lead to errors in the predicted level of traffic. In an attempt to enhance the efficiency of the proposed model (spike ISDN-IoT), we compare it with the deep learning CNN traffic prediction.

III. SYSTEM MODEL

Fig. 1, illustrates the proposed model that is introduced in this paper. The occurred advancement in the science of networks, communications and artificial intelligence have boosted using these technologies in different facet of life. The application of the proposed model in the field of health, specifically, in hospitals in Iraq is our focus. The model consists of a sensing plane, control plane and application plane of an spike ISDN-IoT network.

A. Sensing plane

The proposed model consists of an IoT patient monitoring zone, which is defined as the number of wireless sensing nodes in the sensing area classified according to their activity into three types, as: Forwarding Cluster Head (FCH), which we refer to as the OpenFlow switch; active node; and sleep node. Active member nodes transmit their data to an FCH and in turn, it forwards aggregated data to the sink node as a GATEWAY (GW), the internal components of which are shown in Fig.2. In practice, the GW connects the WSN using a point-to-point connection over the Internet. That is, it can connect to the Internet via local routers with firewalls. [23].In this paper, we propose an intelligent SDN stack for routing and traffic management of patient sensor data. The packet flow that arrives from the buffer of the FCHs with a number of active sensors is destined for the hospital cloud network, as shown in Fig. 1.The FCH approach has two phases: setup and steady-state. In the setup phase, where the FCHs are chosen, each sensor node belongs to its FCH and a cluster is formed, with every node that is not an FCH determining its neighbors and its distance. Secondly, during the steady-state phase, every active sensor begins to send data to its FCH. The FCH approach takes into account some basic factors: residual energy of the sensor nodes, their density and the residual capacity of the buffer size. This is explained in the following equation:

$$IR_N = f(\{EN_N \times \alpha_N \times d_N \quad if \quad d_N \geq d_{th}\}) \tag{1}$$

where, IR_N, EN_N and d_N represent the weight, the residual energy and the density of the sensor N sequentially.$f(\cdot)$ is a nonlinear function which represents the performance of the ANN reactive controller, and d_{th} is the minimum density threshold.The term density of one node is the amount of aggregated neighboring nodes in a place in range r. α_N is the factor of flow buffer size capacity for every sensor as described in the following equation:

$$\alpha_N = \frac{\alpha^N_{max}}{no.\ of\ alive\ sensor\ nodes\ in\ range\ r}. \tag{2}$$

where, α^N_{MAX} is the maximum capacity of flow buffer size in the sensor. Each node manages itself in terms of determining whether to be active and be able to transmit its data or remain in sleep mode. To avoid congestion in the FCHs' flow buffer, which might not have enough capacity to accommodate the sensory-data, the approach has the capability of making the number of active sensors coordinate with their FCHs buffer size. The number of active nodes S_A is determined as in the following equation:

$$S_A = \frac{flow\ table\ size\ of FCH}{total\ rate\ of\ sensor} \tag{3}$$

The proposed FCH approach is used to improve the QoS by reducing packet flow loss and overflow on the FCH flow buffer. The sensor nodes can generate data packets and forwarding data as OpenFlow switches do.

.

B. Control plane

Consider that spike ISDN-IoT is constructed in a homogeneous network, as shown in Fig. 1, consisting of a number of sensors used to sense data from different devices with different types of traffic. The periodic data are collected from a sensor, e.g., the temperature of a patient or blood pressure. In our case, the sensors can collect patient data dynamically to stimulate preventive care, diagnostics etc. and to measure treatment results. The hospital cloud network in Fig. 1 consists of a number of routers, the number depending on the number of considered switches. Each router has its First-come First Served (FCFS) buffer with a predefined capacity. OpenFlow was designed as one of the first SDN standards.It basically defines the communication protocol in SDN environments and enables the SDN controller to combine directly with its data plane.The communication delay between the data plane and control plane is neglected as it is negligible compared to the distance between data plane and cloud.

C. The intelligent SDN stack

SDN technology can work with WSN to verify the activation of sensor nodes in real-time to meet application requirements [22]. The intelligent controllers are the brain of the SDN control layer, which manage the traffic flow of spike ISDN-IoT. We propose an SDN intelligent stack that has two intelligent controllers. These controllers are described as follows:

1) PRSNN Congestion Controller: The structure of PRSNN consists of one input node, a hidden layer with a number of neurons with self-feedback and one output node, as shown in Fig.3.The presence of many hidden layers decreases the speed of the training process and increases network complexity. The PRSNN controls and estimates the packet flow (pf) for the next round in order to reduce the congestion that could occur in the network.

Fig 4 shows the proposed queuing model, where error (t) is the difference between the desired and actual occupancy of the buffer size. The proposed controller is responsible for estimating a suitable amount of packet flow for the next round, with PRSNN training offline to identify the capacity of the buffer size. The total waiting time of the packets in the queue is the sum of the round-trip communication delay in the links and the queuing processing delay in the cloud. To explain the performance of the proposed model, it is taken that we have sensors/switches (IoT patient monitoring zone) to be controlled, as shown in Fig.5. The packet flow is defined as:

$$pf(k+1) = sat[ff(pf(k) + Tu(k)] \tag{4}$$

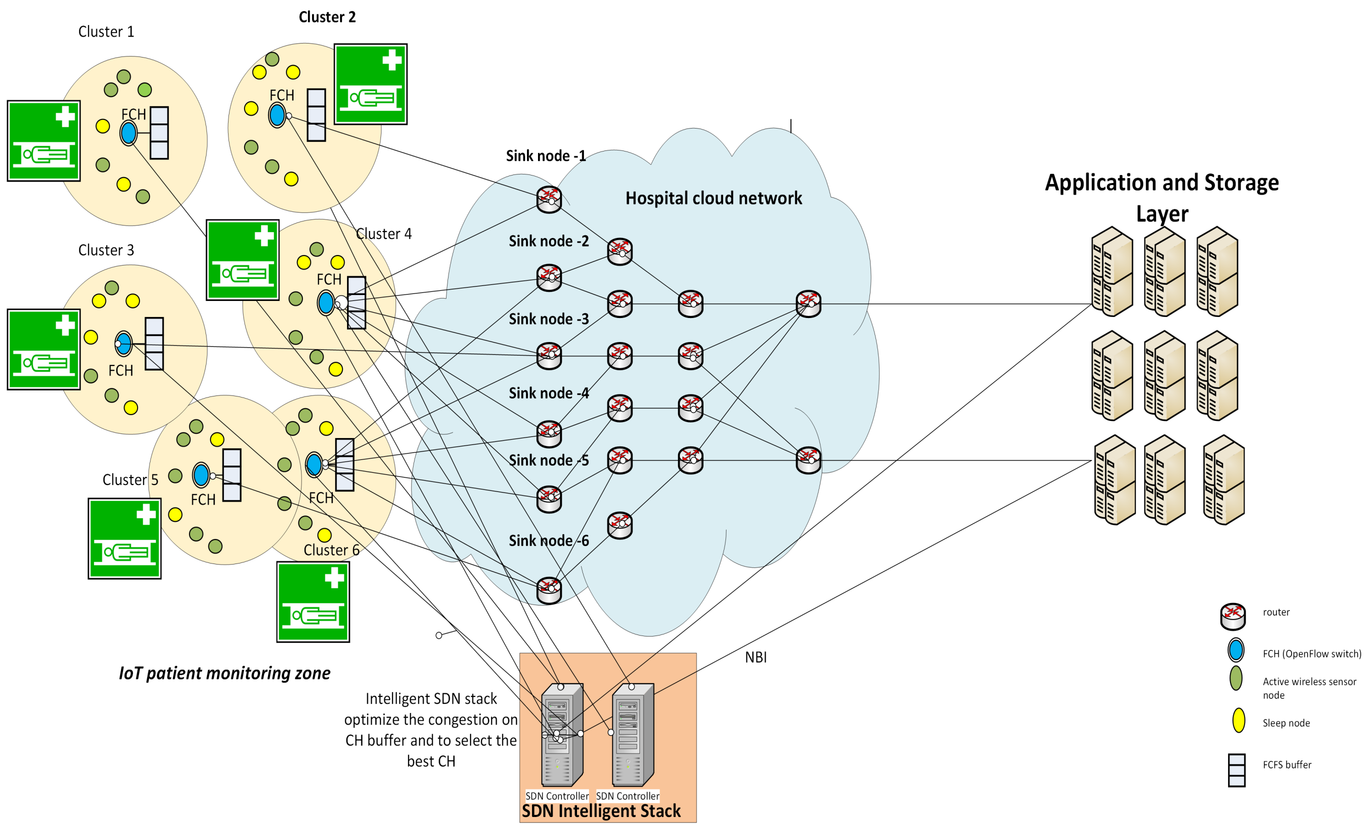


Fig. 1. Proposed Spike ISDN-IoT Network.

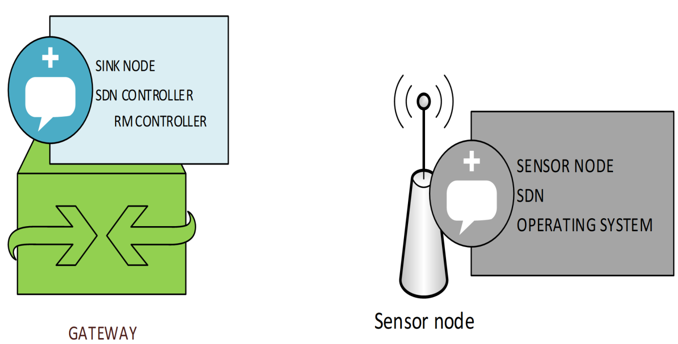


Fig. 2. The internal structure of gateway and sensor node.

Where, $pf(k)$ is the packet flow at time k, T is the sampling period, $u(k)$ is the control law signal and $sat[\cdot]$ is the saturation function. The nonlinear function $ff(\cdot)$ represents the actual packet flow, which is considered as being unknown. The $ff(\cdot)$ is also a function of buffer size, traffic input and available service capacity at the given sensor nodes. The packet flow rate input controller is calculated as:

$$u(k) = \frac{1}{T}(pfd - \hat{f}(pf(k)) + k_v e(k)) \tag{5}$$

where, k_v is the coefficient of the proportional integral controller (PI) used here to increase the accuracy and to eliminate the steady state error as well as keeping the network stable throughout the training process, while $\hat{f}(pf(k))$ is the estimated packet flow and the pfd is the desired packet flow. PRSNN in Fig.5 trains on-line to estimate the packet flow. The minimum rate b_N at the sensor N, is defined as:

$$b_N = Q_M log(R_M) \tag{6}$$

where, Q_M is the size of the queue (buffer) of the (M) FCH node with the corresponding rate R_M. The optimization issue assigns link bandwidth in such a way that the overall spike ISDN-IoT network utilization N_U is maximized as in the following formula:

$$N_U = maximize \sum_M Q_M log(R_M) \tag{7}$$

2) The ANN Controller: The other intelligent controller is based on an ANN (FeedForward Neural Network with one hidden layer), as shown in Fig.6. We are proposing it being used to select the best FCH OpenFlow to carry traffic.The IoT patient monitoring zone is managed based on an ANN, taking the factors described in section III (A) as input to it. While its output is the logical value, where logic 1 is defined as an FCH and logic 0 are cluster members (CM).The back-propagation training algorithm is used to update the weights in an on-line manner.

IV. MODIFIED TRAINING ALGORITHM

In this section, the modified training algorithm used to learn the PRSNN controller is explained. The negative gradient

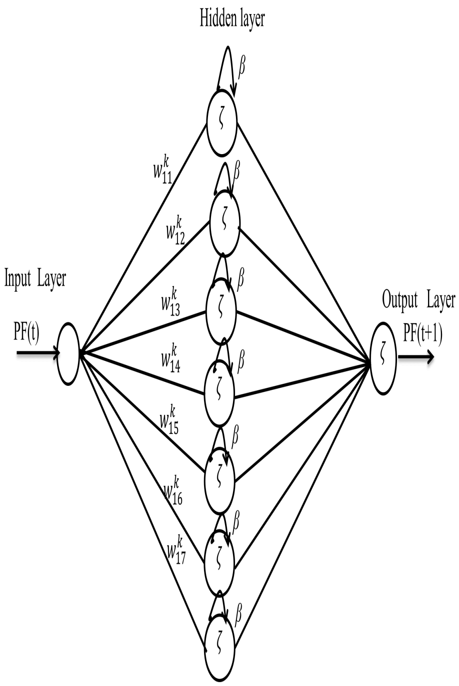


Fig. 3. Structure of the partial recurrent spike neural network.

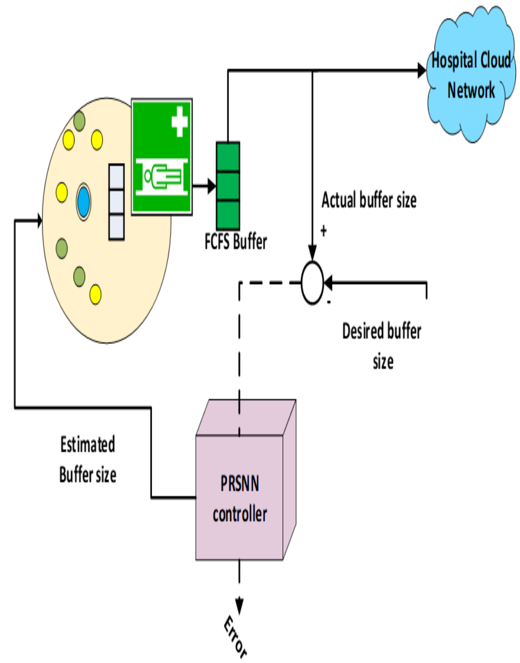


Fig. 4. The proposed queuing model.

descent approach for minimizing the difference between the desired and actual packet flow and the modified spiking algorithm [31] are the core of the proposed algorithm.

The internal connection single synaptic of PRSNN is shown in Fig. 7 a and the broken line portion of single synaptic terminal in Fig.7 b. represents a time delayed synaptic connection between two neurons. In the Fig.7 b. the neuron i is not permitted to spike anymore through the resting period of T time interval, when the threshold value θ has been overstepped at a specific instant t_i and it will be reset in the next, $t_i + d^k$. The whole single connection amidst the layers in PRSNN is constructed of a class with the same number of synaptic terminals. It is clear from the Fig.7 a that each sub-connection is having a different weight and delay. The difference between the time of the postsynaptic potential and the firing of presynaptic neurons i can be identified as the delay of the synaptic terminals. The time of postsynaptic potential starts to grow, as seen in Fig.7b, and there is a synapse chain in the connection. The spike-response function ζ is affected by the weight of each synapse. The input of PRSNN is assigned to the packet flow accumulation rate $pf(t)$, i.e., the number of flow packets arriving at the SDN controller from the network.The parameters that are trained in the proposed algorithm are the weights, threshold, and synaptic delays. The number of synapses between the input and hidden layers as well as between the hidden and output layers is updated. This number is generally chosen analytically at the initial phase.At the beginning, the weights are initiated randomly between [-0.5,+0.5] and then, after implementing epochs of training, the weight values and the learning rate η are adapted more efficiently.

The desired and the actual packet flows are at first encoded into spike times as demonstrated in the equation below:

$$t_h^f = t_{max} - \lfloor \frac{t_{min}(pf(t) - pf_{min})(t_{max} - t_{min})}{(pf_{max} - pf_{min})} \rceil. \quad (8)$$

where, pf_{max} and pf_{min} represent the maximum and minimum real flow, whilst t_{max} and t_{min} are the maximum and minimum interval time, respectively. The function $\lfloor \rceil$ is a round function.

The flow packet decoding is explained in the equation:

$$pf(t_j) = \frac{(t_{max} - t_j - t_{min})(pf_{max} - pf_{min})}{(t_{max} - t_{min})} + pf_{min}. \quad (9)$$

In the training algorithm, there are two phases. The feed-forward phase, where each neuron spikes at each time interval T only once at most. This happens when the value of threshold θ is overstepped the membrane potential m . The feed-forward phase begins from the hidden layer I with neuron (i) being

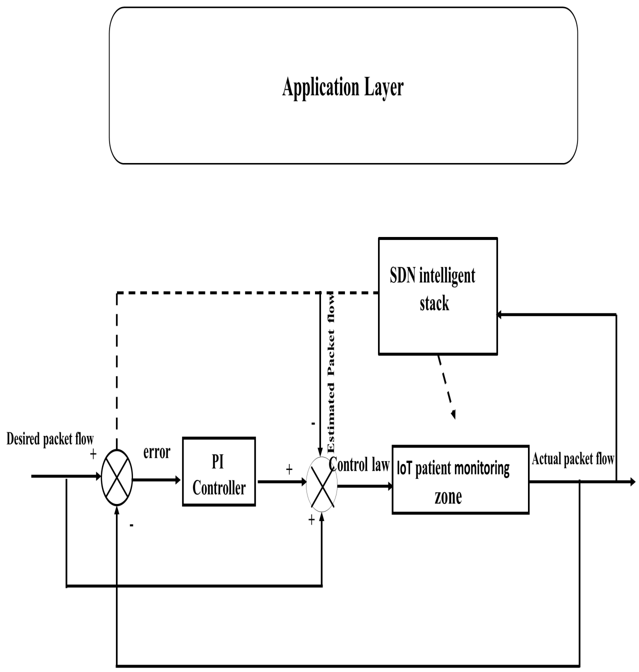


Fig. 5. The structure of the proposed congestion control.

continuously examined to see whether it is spiked or not. When the neuron (i) is spiked,the algorithm uses the next neuron $(i+1)$. The membrane potential $m_i(t)$ is computed by the training algorithm ,based on (10), according to input spikes t_h^f of neuron h at the input layer.

$$\begin{aligned} m_i(t) &= \sum_{h=1}^{NH}\sum_{k=1}^{D} w_{hi}^k \zeta(t - t_h^f - d^k) \\ &+\beta * \sum_{h=1}^{NH}\sum_{k=1}^{D} w_{hi}^k * pf_{hi}^k(t-1). \end{aligned} \tag{10}$$

The self-feedback β in PRSNN structure is a constant value between (0-1). The term $pf_{hi}^k(t-1)$ means the past packet flow as the input to the PRSNN. The activation function $\zeta(t - t_h^f - d^k)$ is computed as:

$$\zeta(t - t_h^f - d^k) = -\sigma * exp\frac{-(t - t_h^f - d^k)}{\tau}. \tag{11}$$

The output layer J will have the same process, which is when the second layer's neurons have finished, the back-propagation phase starts.

The synapse weights of connection are updated when the feed-forward phase has finished. Different to feed-forward, back-propagation starts from the output layer and comes back to the hidden layer. For clarification, we defined the function $\zeta(t - t_h^f - d^k)$ as y_h^k and $\zeta(t - t_i^f - d^k)$ as y_i^k. The error E

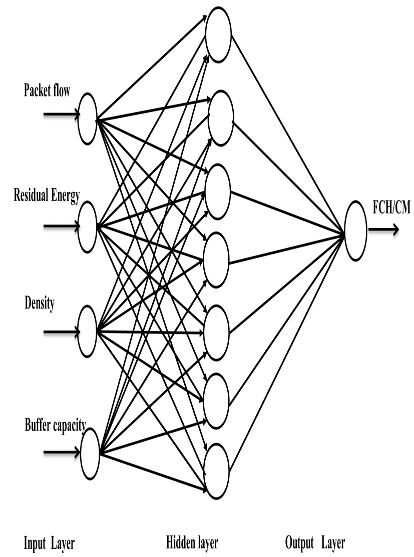


Fig. 6. The structure of the artificial neural network selection process .

which is defined as the difference between the target and real spike time of the neuron is expressed as:

$$E = (T_j - t_j^f). \tag{12}$$

The synapses of the hidden layer and output layer will be updated according to (13-18).

$$w_{ij}^k(t+1) = w_{ij}^k(t) - \Delta w_{ij}^k(t). \tag{13}$$

where,

$$\Delta w_{ij}^k(t) = \eta.\delta_j.y_h^k. \tag{14}$$

$$\delta_j = \frac{E}{\sum_{(i=1)}^{In}\sum_{(k=1)}^{D} w_{ij}^k \frac{\partial y_i^k}{\partial t}}. \tag{15}$$

$$\delta_i = \frac{\sum_{(i=1)}^{(In)} \delta_j \sum_{(k=1)}^{D} w_{ij}^k \frac{\partial y_i^k}{\partial t}}{\sum_{(i=1)}^{Hn}\sum_{(k=1)}^{D} w_{hi}^k \frac{\partial y_h^k}{\partial t}}. \tag{16}$$

$$w_{hi}^k(t+1) = w_{hi}^k(t) - \Delta w_{hi}^k(t). \tag{17}$$

where,

$$\Delta w_{hi}^k(t) = \eta.\delta_i.y_i^k. \tag{18}$$

The synaptic delay and neuron thresholds updating are defined in the following formulas:

$$\Delta_{hi}^k = -\rho_d \sum_{(i=1)}^{(NI)} \frac{\partial E}{\partial t_j^f}\frac{\partial t_j^f}{\partial y_h^k(t)}\frac{\partial y_h^k(t)}{\partial d_{hi}^k}|_{(t=T_j)}. \tag{19}$$

TABLE I

Parameters of the partial recurrent spike neural network training algorithm

Symbol	Meaning
σ	Constant of the activation function
η	Learning rate
θ	The threshold value
ρ_d	Learning rate of the synaptic delay
ρ_θ	Learning rate of the synaptic thresholds
τ	The time constant
δ	The delta function
d^k	delay of the connection
m_i	Membrane potential of neuron i at the hidden layer
m_j	Membrane potential of neuron j at the output layer
w_{hi}^k	Sub-connection weight between the input and hidden layers
w_{ij}^k	Sub-connection weight between the hidden and output layers
Δt	Step time
D	Number of delayed synapses per connection
H	Input layer
I	Hidden layer
J	Output layer
T_j	Target spike time of the output neuron
t_j^f	The real spike time of output neuron
NH	Number of neurons in the input layer
NI	Number of neurons in the hidden layer
y_h^k	The output of the hidden layer
y_i^k	The output of the output layer
T	Time interval
max. epoch	Maximum number of epochs
h	Neuron sequence in the input layer
i	Neuron sequence in the hidden layer
j	Neuron sequence in the output layer

$$\Delta\theta_j = -\rho_\theta \sum_{(i=1)}^{(NI)} \frac{\partial E}{\partial t_j^f} \frac{\partial t_j^f}{\partial y_h^k(t)} \frac{\partial y_h^k(t)}{\partial \theta_j} |_{(t=T_j)}. \tag{20}$$

Table I explains all the symbols and parameters of equations.The parameters are updated in the training algorithm with the initial values are chosen by trial and error. PRSNN is adaptive according to the traffic dynamics and the data plane performance, such that the proactive controller keeps a balance between the buffer sizes and traffic flow of the network.PRSNN achieves both data plane efficiency (high traffic flow rate) and stability. The flow chart of the proposed model is shown in Fig. 8 and the training algorithm of PRSNN is shown in Figs. 9 and 10.

V. EVALUATION SETUP

We consider scenarios with N sensors that are placed in a random way in a sensing square area of (150× 150) meters, with the transmission range of each sensor being fixed at 25m. We vary the number of sensors (80 and 120) to control the density of the network and the implementation for the area is shown in Fig.11. The sensors generate traffic at the beginning of each scheduling period.That is, they implement low to high flow and then, this traffic is routed to the FCH. The PRSNN controller contributes to minimizing the congestion level. That

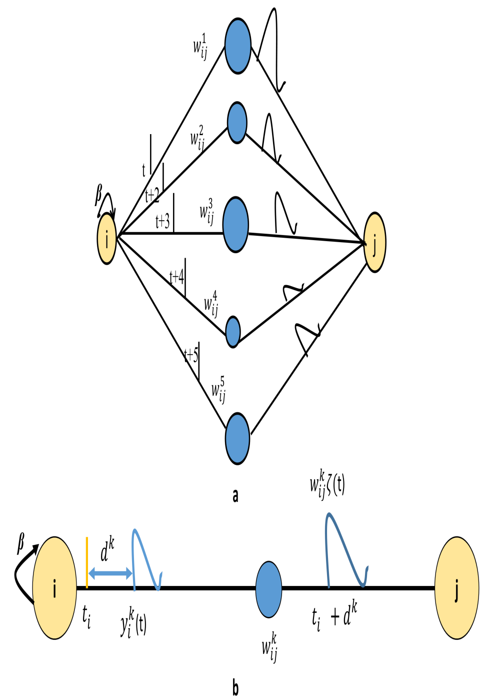


Fig. 7. a: Internal connection single synaptic of the PRSNN. b: Single synaptic terminal.

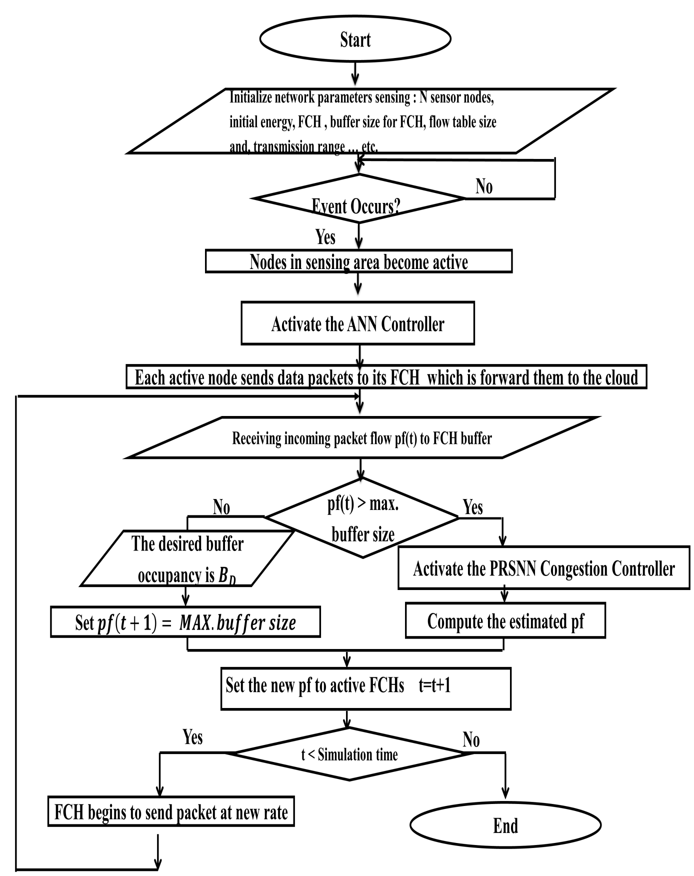


Fig. 8. Flowchart of the proposed model.

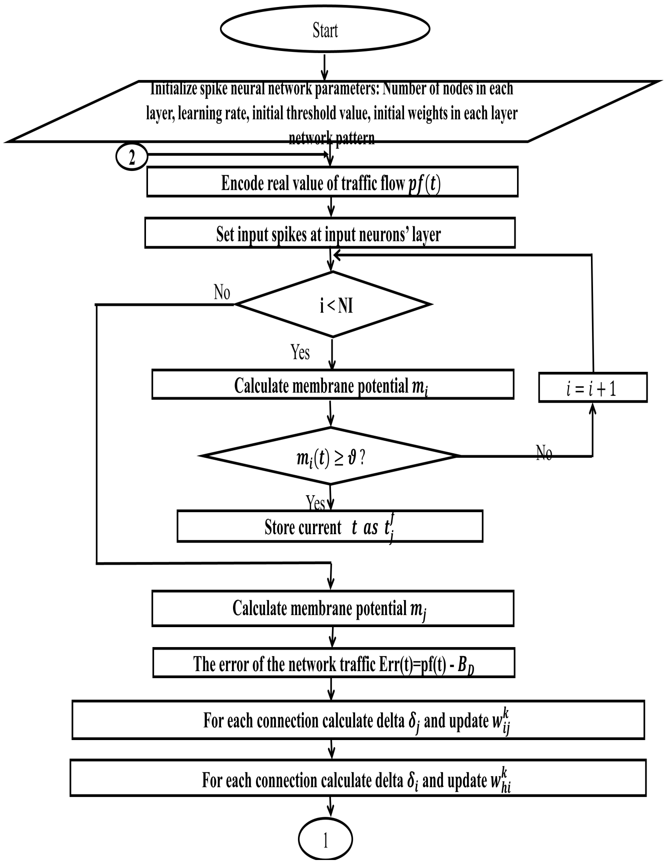


Fig. 9. The proposed training algorithm.

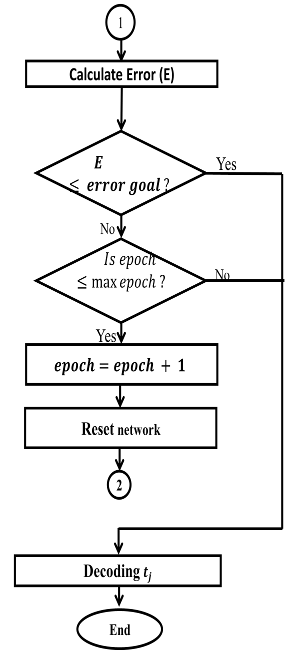


Fig. 10. Continue:The proposed training algorithm.

TABLE II
PARAMETERS OF THE SIMULATION

Coverage area	150 meters × 150 meters
Number of nodes	80,120
Buffer size of FCH	250 packets
Buffer size of each sensor node	50-100 packets
Data packet size	800 byte
Simulation time	250 msec
Data packet generating for each node	5(packet/msec.)

is, the FCHs are classified as congestion, if this percentage exceeds a threshold level. In this paper, the threshold level is set at 90% of the queue buffer size and it is selected based on experiential evaluation.

The simulation is run with the parameters described in Table II and with the Python programming language and Mininet simulator.

The following assumptions are applied for the network:

1. All stationary active sensor nodes generate static flow per unit of time;

2. There are two activities for the sensor node, the first being to generate flow traffic and the second is forwarding this traffic to the FCH;

3. The connection between the cloud, FCH and its member nodes comprises bidirectional single hop wireless links with an OpenFlow SDN switch;

4. Sensor nodes can verify their mode according to the FCH buffer capacity and its density;

5.The amount of flow (traffic generated) sent by the sensor node must be within the capacity of the channel of the network.

To show the efficiency of the proposed model, a comparison is made between the it and that with a controller based on CNN. Fig. 12 shows the structure of CNN for a controller with one convolutional layer, a ReLU layer, and a fully connected layer used for the estimated traffic in a spike ISDN-IoT network. The reason behind choosing CNN to compare with it, is that, it is more efficient than the traditional neural network, as explained in section II on related work.

The input of the CNN will be the features of the traffic flows, including the packet generation rate of every FCH,lengths of the packet queues in the buffers of the FCHs. The output is collected as two binary values, which when set at (1,0) shows that the path mixture will lead to congestion and otherwise (i.e., 0,1), it will not. Clearly, the path mixtures that will not lead to congestion will be chosen. The CNNs will be periodically updated, while they are being used to select the path mixture. Every FCH will keep listing its traffic flow and then send the data to the SDN controller. The controller uses the data for the purpose that the traffic patterns of all FCH will be arranged in a matrix and then used as the input of the CNNs to choose the path mixture for the next time interval.

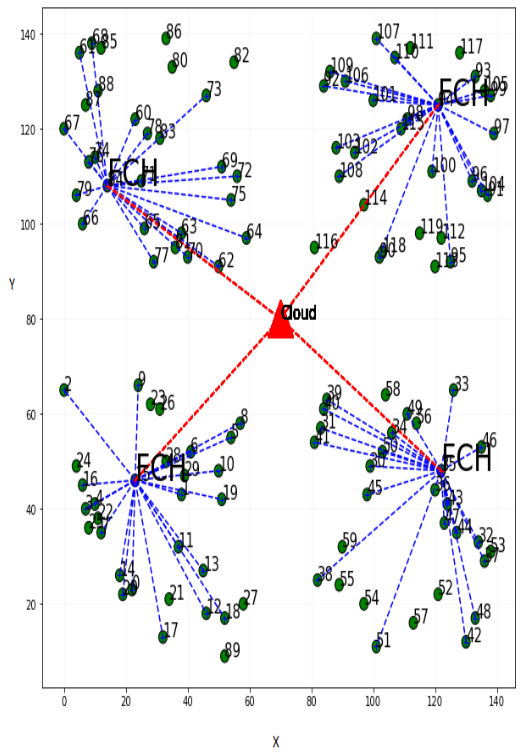


Fig. 11. The simulation area with 120 sensors nodes.

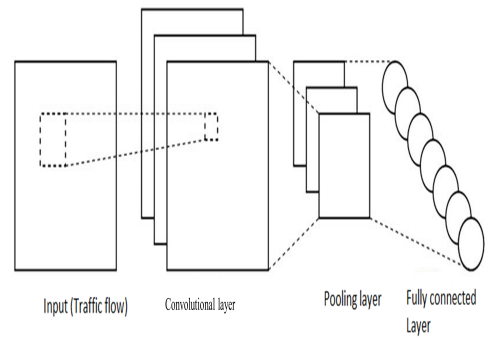


Fig. 12. The Convolutional Neural Network model.

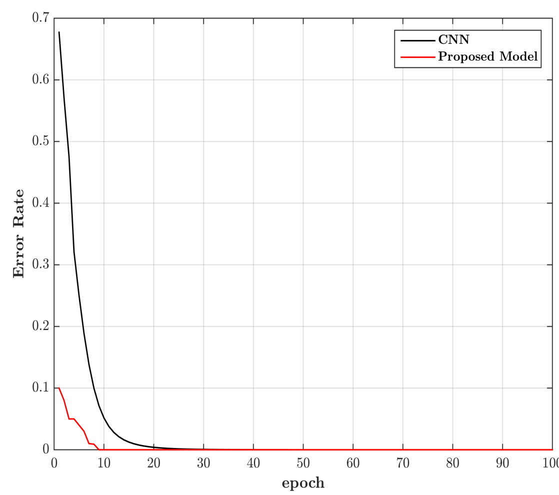


Fig. 13. The minimization of error during training.

Fig. 13 shows the minimization of error during the training process. It is clear from the Fig. 13 that PRSNN can reach to the error goal, which is set to 10^-5, faster than CNN. This is because not all the neurons will update their weights all the time, but just those that exceed the threshold value will be spike. So, the modified training algorithm which we propose to train PRSNN is more powerful than the back-propagation training algorithm used to train CNN.

Fig.14 shows a comparison of the actual and estimated pf forwarded by the network and when the number of sensor nodes is 80. It can be seen that the performance of the proposed model is better than CNN, which is very clear when the network keeps its traffic with a buffer capacity size of FCH. In this simulation, we have four FCHs. When all are active, the network with the proposed model and CNN can operate in high traffic flow, thereby controlling the traffic in order to mitigate congestion at the buffer. The proposed model has a better ability at estimating the packet flow than with CNN. This is because the training algorithm can enhance the performance of PRSNN. It works with a high capability of estimation of the rate of packet flows. Fig. 15 illustrates the performance of the proposed model and CNN when the number of sensor nodes is increased to 120.Thus, the proposed model can work as accurately as CNN compared with the CNN the proposed model can still work accurately. In sum, the proposed congestion controller in the spike ISDN-IoT control plane is able to process all the requests coming from the switches even when the number is increased.

VI. Performance Metrics

The performance of the proposed model, and CNN are explained with respect to QoS in terms of Packet Loss Ratio (PLR), Network Energy Consumption (NEC), Buffer Utilization Ratio (BUR), Network Throughput Ratio (NTR), and Network Lifetime (NLT).

A. Packet Loss Ratio (PLR)

Fig. 16 presents the PLR in the spike ISDN-IoT network, when the proposed model is implemented. In Fig. 16, a

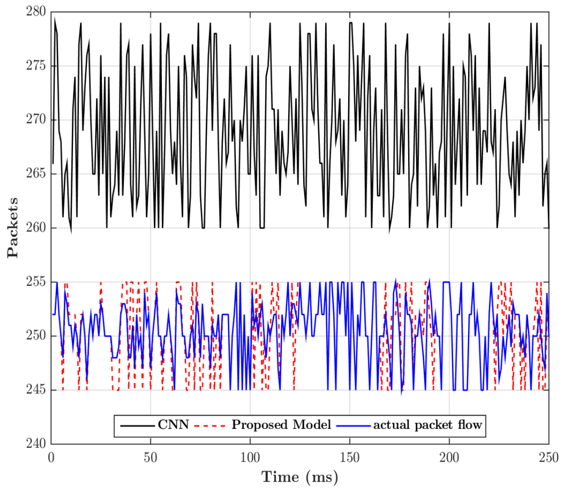


Fig. 14. Comparison of the estimated PF between the proposed model and CNN when the number of sensor nodes is 80

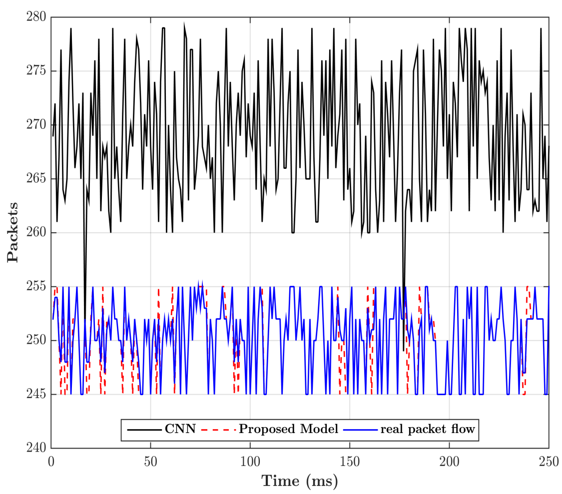


Fig. 15. Comparison of the estimated PF between the proposed model and CNN when the number of sensor nodes is 120

comparison between the proposed model and CNN when the number of sensor nodes is 80 is provided. We can observe from the figure that the PLR of the proposed model is better than that for the CNN, because the congestion controller is able to decrease the sending rate of the active clusters during the transmission process. It is also clear that whilst the CNN performs well, it is not as accurate as the proposed model. This means that, the proposed intelligent queuing model has good ability to estimate the capacity of the buffer size in the network and manage the queue of the packet flow accurately.

B. Network Energy Consumption (NEC)

Fig. 17 compares the energy consumption of FCH in the network for the proposed model and CNN, with respect to time, when the number of sensor nodes is 80. The result of the comparison demonstrates that the network energy consumption with the proposed model is better than that with CNN. Thus,

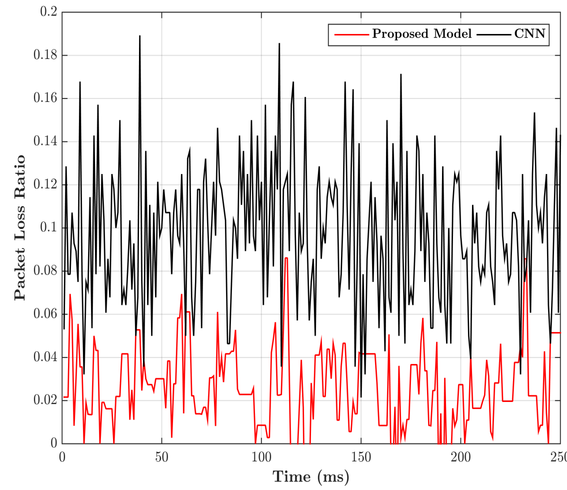


Fig. 16. Comparison of the packet loss ratio between the proposed model and CNN when the number of sensor nodes is 80.

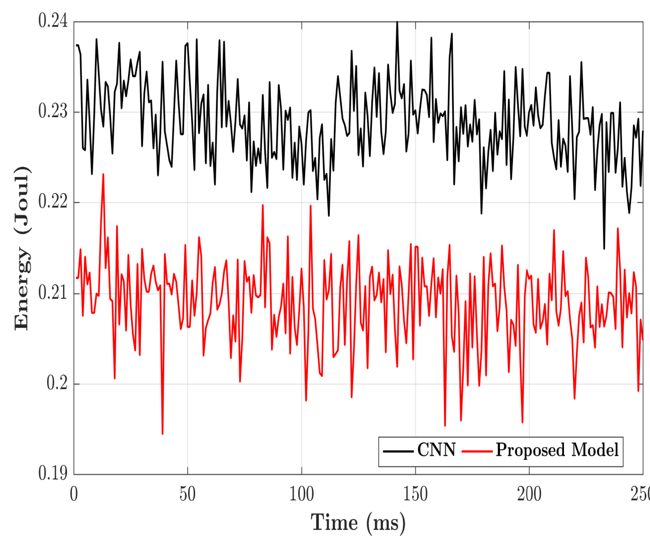


Fig. 17. Comparison of the network energy consumption between the proposed model and CNN when the number of sensor nodes is 80.

the proposed model can decrease the energy consumed in dropped packets by overflow to an acceptable value. In the proposed training algorithm, not all the neurons are firing; just those that have reached threshold value. This means that the proposed model does not need as much time for training as with CNN. Also, separating the sensing area in the spike ISDN-IoT network into a number of FCHs, based on an ANN controller, provides the capability of minimizing the energy consumption of the whole network.

C. Buffer Utilization Ratio (BUR)

Fig 18 denotes the buffer utilization ratio of the network using the proposed model compared with that for CNN, when the number of sensor nodes deployed in spike ISDN-IoT is 80. It is clear that the controlled network guarantees a better buffer utilization ratio than for CNN. Clearly, the proposed

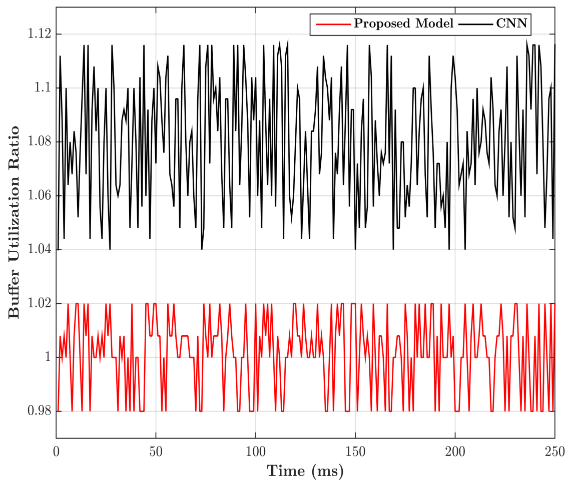


Fig. 18. The buffer utilization ratio when the number of sensor nodes is 80

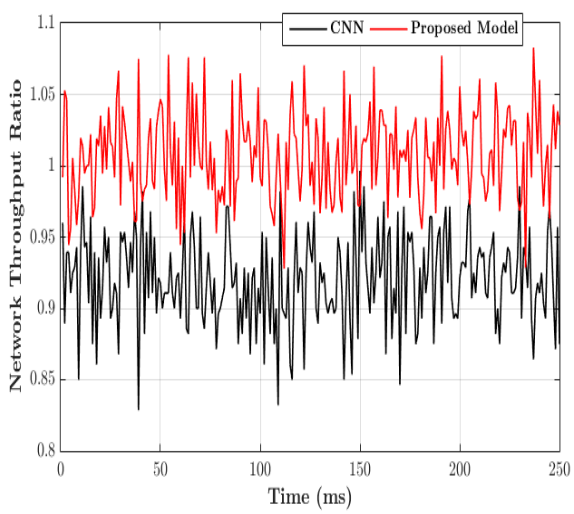


Fig. 19. The network throughput ratio when the number of sensor nodes is 80

model performs well with high accuracy, much more so than with CNN. The idea behind using the PRSNN as congestion controller is to increase the power of the network in estimating the packet flow. The strength of PRSNN is acquired from accurate modeling of the synaptic interactions between the biological neurons, taking into consideration the time of spike firing. The PRSNN computational power, thus, exceeds that of CNN which uses sigmoidal or wavelet activation functions. Furthermore, PRSNN has the ability for swift adaptation.

D. Network Throughput Ratio (NTR)

The NTR is defined as the proportion of the received packets by the gateway over the total number of packets generated by the FCH during the simulation time. Fig. 19. display a comparison between the proposed model and the CNN, when numbers of sensor nodes is 80. It is clear from the figure that the proposed model outperforms CNN, with a higher throughput ratio.The spike ISDN-IoT network with the proposed model is able to keep the throughput ratio to 100%, whereas CNN cannot.In the proposed model, all the parameters (which have been described in section III) that have a positive effect on the performance of the network, have been taken into consideration.The performance of the SDN intelligent stack in our proposed model can efficiently manage the traffic load.

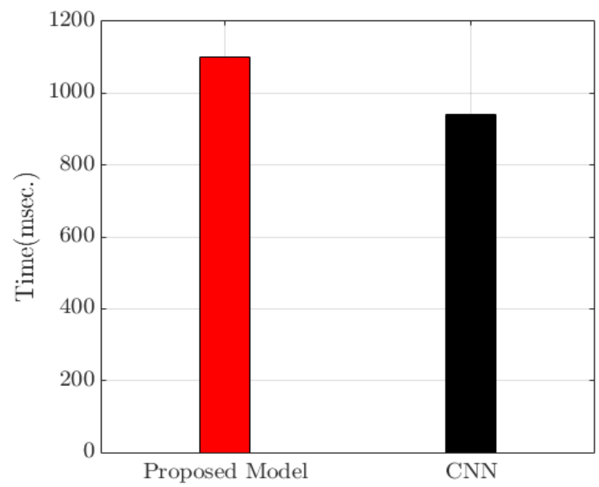


Fig. 20. The network lifetime

E. Network Lifetime (NLT)

This refers to the time required to drain the energy of all the sensors nodes in the network. Fig. 20. shows a comparison of NLT when the proposed model and CNN are used. It is clear that the proposed model prolongs it more than CNN. The concept of FCHs introduced in this paper with an ANN controller successfully increases the lifetime of the network, which means that the sensors can keep their energy for a longer time than with other methods, like CNN.

VII. CONCLUSION

In this paper, we have proposed spike ISDN-IoT architecture for utilization in health care applications. We have proposed two intelligent controllers in the SDN intelligent stack, which has the capability of estimating the packet flow of the sensing area. One of the proposed controllers works proactively in a Partial Recurrent Spike Neural Network to estimate the packet flow of the sensing area.The other works as a reactive one based on an ANN, being tasked with selecting the cluster head and its members. The simulation results have proven that the QoS is enhanced in the spike ISDN-IoT network. The ANN controller delivers the capability of selecting the cluster head and its members efficiently in the sensing area, which is clearly shown in the results for QoS. The packet flow rate is estimated by the proposed model, which coordinates the available capacity of the buffer with a number of active sensor nodes in the network to prevent buffer overflow. Controlling the network by the proposed model has more accuracy than with CNN, which is because of the spiking power of the proposed training algorithm.

A. Acknowledgment

This work is supported by Ministry of Higher Education and Scientific Research, University of Baghdad, Iraq.

References

[1] S. Bera, S. Misra, S. K. Roy, and M. S. Obaidat, “Soft-wsn: Software-defined wsn management system for iot applications,” *IEEE Systems Journal*, vol. 12, no. 3, pp. 2074–2081, 2016.

[2] A. Ghosh, D. Chakraborty, and A. Law, “Artificial intelligence in internet of things,” *CAAI Transactions on Intelligence Technology*, vol. 3, no. 4, pp. 208–218, 2018.

[3] D. Kaur, G. S. Aujla, N. Kumar, A. Y. Zomaya, C. Perera, and R. Ranjan, “Tensor-based big data management scheme for dimensionality reduction problem in smart grid systems: Sdn perspective,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 10, pp. 1985–1998, 2018.

[4] Z. Han, T. Lei, Z. Lu, X. Wen, W. Zheng, and L. Guo, “Artificial intelligence based handoff management for dense wlans: A deep reinforcement learning approach,” *IEEE Access*, 2019.

[5] R. Thupae, B. Isong, N. Gasela, and A. M. Abu-Mahfouz, “Machine learning techniques for traffic identification and classifiacation in sd-wsn: A survey,” in *IECON 2018-44th Annual Conference of the IEEE Industrial Electronics Society*. IEEE, 2018, pp. 4645–4650.

[6] Q. Li, N. Huang, D. Wang, X. Li, Y. Jiang, and Z. Song, “Hqtimer: A hybrid $\{Q\}-$ learning-based timeout mechanism in software-defined networks,” *IEEE Transactions on Network and Service Management*, vol. 16, no. 1, pp. 153–166, 2019.

[7] J. Xie, F. R. Yu, T. Huang, R. Xie, J. Liu, C. Wang, and Y. Liu, “A survey of machine learning techniques applied to software defined networking (sdn): Research issues and challenges,” *IEEE Communications Surveys & Tutorials*, vol. 21, no. 1, pp. 393–430, 2018.

[8] A. Rego, A. Canovas, J. M. Jiménez, and J. Lloret, “An intelligent system for video surveillance in iot environments,” *IEEE Access*, vol. 6, pp. 31 580–31 598, 2018.

[9] F. Tang, B. Mao, Z. M. Fadlullah, and N. Kato, “On a novel deep-learning-based intelligent partially overlapping channel assignment in sdn-iot,” *IEEE Communications Magazine*, vol. 56, no. 9, pp. 80–86, 2018.

[10] C. Yu, J. Lan, Z. Guo, and Y. Hu, “Drom: Optimizing the routing in software-defined networks with deep reinforcement learning,” *IEEE Access*, vol. 6, pp. 64 533–64 539, 2018.

[11] X. Huang, T. Yuan, G. Qiao, and Y. Ren, “Deep reinforcement learning for multimedia traffic control in software defined networking,” *IEEE Network*, vol. 32, no. 6, pp. 35–41, 2018.

[12] F. Tang, Z. M. Fadlullah, B. Mao, and N. Kato, “An intelligent traffic load prediction-based adaptive channel assignment algorithm in sdn-iot: A deep learning approach,” *IEEE Internet of Things Journal*, vol. 5, no. 6, pp. 5141–5154, 2018.

[13] B. Mao, F. Tang, Z. M. Fadlullah, and N. Kato, “An intelligent route computation approach based on real-time deep learning strategy for software defined communication systems,” *IEEE Transactions on Emerging Topics in Computing*, 2019.

[14] B. Mao, F. Tang, Z. M. Fadlullah, N. Kato, O. Akashi, T. Inoue, and K. Mizutani, “A novel non-supervised deep-learning-based network traffic control method for software defined wireless networks,” *IEEE Wireless Communications*, vol. 25, no. 4, pp. 74–81, 2018.

[15] G. Li, X. Wang, and Z. Zhang, “Sdn-based load balancing scheme for multi-controller deployment,” *IEEE Access*, vol. 7, pp. 39 612–39 622, 2019.

[16] A. Tavanaei, M. Ghodrati, S. R. Kheradpisheh, T. Masquelier, and A. Maida, “Deep learning in spiking neural networks,” *Neural Networks*, 2018.

[17] H. Mostafa, “Supervised learning based on temporal coding in spiking neural networks,” *IEEE transactions on neural networks and learning systems*, vol. 29, no. 7, pp. 3227–3235, 2017.

[18] N. Kumar and D. P. Vidyarthi, “A green routing algorithm for iot-enabled software defined wireless sensor network,” *IEEE Sensors Journal*, vol. 18, no. 22, pp. 9449–9460, 2018.

[19] F. P.-C. Lin and Z. Tsai, “Hierarchical edge-cloud sdn controller system with optimal adaptive resource allocation for load-balancing,” *IEEE Systems Journal*, 2019.

[20] Y. Xu, M. Cello, I.-C. Wang, A. Walid, G. Wilfong, C. H.-P. Wen, M. Marchese, and H. J. Chao, “Dynamic switch migration in distributed software-defined networks to achieve controller load balance,” *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 3, pp. 515–529, 2019.

[21] I. Haque, M. Nurujjaman, J. Harms, and N. Abu-Ghazaleh, “Sdsense: An agile and flexible sdn-based framework for wireless sensor networks,” *IEEE Transactions on Vehicular Technology*, vol. 68, no. 2, pp. 1866–1876, 2018.

[22] S. Misra, S. Bera, M. Achuthananda, S. K. Pal, and M. S. Obaidat, “Situation-aware protocol switching in software-defined wireless sensor network systems,” *IEEE Systems Journal*, vol. 12, no. 3, pp. 2353–2360, 2017.

[23] G. M. Dias, C. B. Margi, F. C. de Oliveira, and B. Bellalta, “Cloud-empowered, self-managing wireless sensor networks: Interconnecting management operations at the application layer,” *IEEE Consumer Electronics Magazine*, vol. 8, no. 1, pp. 55–60, 2018.

[24] J. Son and R. Buyya, “Priority-aware vm allocation and network bandwidth provisioning in software-defined networking (sdn)-enabled clouds,” *IEEE Transactions on Sustainable Computing*, 2018.

[25] B. K. Al-Shammari, N. Al-Aboody, and H. S. Al-Raweshidy, “Iot traffic management and integration in the qos supported network,” *IEEE Internet of Things Journal*, vol. 5, no. 1, pp. 352–370, 2017.

[26] A. Taherkhani, A. Belatreche, Y. Li, and L. P. Maguire, “A supervised learning algorithm for learning precise timing of multiple spikes in multilayer spiking neural networks,” *IEEE transactions on neural networks and learning systems*, no. 99, pp. 1–14, 2018.

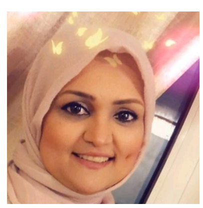
[27] Y. Miao, H. Tang, and G. Pan, “A supervised multi-spike learning algorithm for spiking neural networks,” in *2018 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2018, pp. 1–7.

[28] F. Han, R. Li, and D. Qian, “Short-term wind speed forecasting model based on spiking neural network,” in *2018 International Conference on Advanced Mechatronic Systems (ICAMechS)*. IEEE, 2018, pp. 359–363.

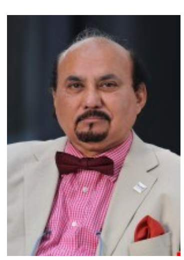
[29] J. H. Lee, T. Delbruck, and M. Pfeiffer, “Training deep spiking neural networks using backpropagation,” *Frontiers in neuroscience*, vol. 10, p. 508, 2016.

[30] S. Matsuda, “Bpspike: a backpropagation learning for all parameters in spiking neural networks with multiple layers and multiple spikes,” in *2016 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2016, pp. 293–298.

[31] N. A. S. Al-Jamali and H. S. Al-Raweshidy, “Modified elman spike neural network for identification and control of dynamic system,” *IEEE Access*, vol. 8, pp. 61 246–61 254, 2020.



N adia A. SH. AL-Jamali (M’10) received a B.Sc. degree in control and systems engineering, M.Sc. degree in control engineering, and Ph.D. degree in computer engineering from the University of Technology, Baghdad, Iraq. She is currently working at Brunel University London, London, U.K. Her fields of interest are computer control, wireless sensor networks, intelligent systems, neural networks and robotics.



H amed S. Al-Raweshidy(SM’03) received B.Eng. and M.Sc. degrees from the University of Technology, Baghdad, Iraq, in 1977 and 1980, respectively, a Postgraduate Diploma from Glasgow University, Glasgow, U.K., in 1987, and a Ph.D. degree from the University of Strathclyde, Glasgow, in 1991, all in electronic engineering. He has worked with the Space and Astronomy Research Center, Baghdad, Perkin Elmer, Waltham, British Telecom, Oxford University, Manchester Metropolitan University, and Kent University, Canterbury, U.K. He is currently the Director of the Wireless Network and Communications Centre, Brunel University London, London, U.K.